MULTIMEDIA UNIVERSITY®

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, **2016/2017**

HPB2021 – BIOINFORMATICS PROGRAMMING II

(All sections / Groups)

19 Oct 2016
09:00-11:00 AM
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 7 pages including cover page.
2. Question one is MANDATORY. Attempt THREE out of FOUR remaining questions. All questions carry equal marks (10), and the distribution of the marks for each question is given.
3. Please write all your answers in the Answer Booklet provided.

QUESTION 1 Mandatory [10 marks]

Write down the expected output of the following codes.

Python

i. [1 mark]

```
def fun(x, y):
    print x + y,
    print x * y

fun("2", 3)
fun("2", "3")
```

ii. [0.5 mark]

```
def what(n):
    if n == 0:
        return 0
    else:
        return n + what(n-1)
print what(5)
```

iii. [1 mark]

```
s = "Alabama and Alaska"
s[ : 7]
s[ 1 : 7 : 2]
s[ 2 : 7 :-1]
s[ : : -1]
```

iv. [1 mark]

```
text = raw input("Input text: ")
vowels = ['a', 'e', 'i', 'o', 'u', 'y' ]
for c in text.lower():
    if not c.isalpha():
        continue
    if c in vowels:
        print c,
        continue
    print "%sO%s"%(c, c),
```

Input text: The light was yellow, sir.

v. [1 mark]

```
m = [ [0] * 3 ] * 3
print m
m[0][2] = 7
print m
```

vi. [0.5 mark]

```
x= "Hello "
y= "World! "
x, y = y, x
print x+y
```

vii. [1 mark]

```
fred = [ 'Alice', 'goes', 'to', 'market' ]
fred.extend([ 'with', 'Mike' ])
last = fred.pop()
fred.append('Fred')
print 'A:', fred
fred = [ 'On', 'Tuesday,' ] + fred
print 'B:', fred
fred.reverse()
print 'C:', fred
fred.sort()
print 'D:', fred
```

viii. [0.5 mark]

```
for y in range(25, 20, -1):
    if y != 25:
        print y,
```

R Programming

i. [1 mark]

```
> x <- c(1:10)
> x[(x>8) | (x<5)]
> x > 8 | x < 5
```

ii. [1mark]

```
> smoker=matrix(c(70,50,65,140),
ncol=2,byro=FALSE)
> rownames(smoker)=c("male","female")
```

	> colnames(smoker)=c("smoke","nosmoke") > smoker= as.table(smokers) > smokers
iii.> seq(3,30,length=10) [1 mark]	iv. > x=1:6 [0.5 mark] > y=cut(x,3) >y

Continued...

QUESTION 2 [10 marks]

a) Imagine that the string 'g' is a DNA sequence from some person's genome. You want to check whether this piece of DNA contains the trinucleotide repeat CAG and, if it does, with what multiplicity. E.g. the repeat CAGCAGCAG has multiplicity 3 since the pattern is repeated 3 times. Complete the given Python program to go through g and find the maximal number of consecutive occurrences of the pattern CAG. Finally print the maximal number.

Suggested output:
Pattern found with max multiplicity 24 [6 marks]

Hint: (With pen and paper, you probably scan the DNA string left to right to spot any CAG repeats, remembering the multiplicity of the longest one seen so far. Do the same thing over and over again for the entire length of the string. Something along these lines, perhaps:
While it is not done, search the string left to right:
1. If you don't see a CAG, move one character down the string.
2. If you see a CAG, you are in a pattern repeat. Add one to the multiplicy of this current repeat and check if it beats the current max.)

```
# NB. The human chromosome 19 is about 80.000 times the
length of this string
g =
    ""CAGTACTACCTCAGACGTCAGCATCAGTACATCGATCGATCGATGCT
    AGCTAGCGCATCGATCTACGTACGTGACTGATCGTACGTACGACATCG
    TCCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAGCAG
    CAGCAGCAGCAGCAGCAGCAGCAGTACGTACGTATCATCTATTAG
    CGCGCGTAATCCCGATATCTTATACTACTCAGGATGCAGCAGCAGCAG
    CAGCAGCAGTGG"""
pointer = 0 # index pointer into g
mul = 0 # multiplicity of current pattern match
maxmul = 0 # max multiplicity of any pattern match found
### Write your code here
.....
###
```

b) Define a function "dig(-)" that receives a sequence and one restriction pattern and returns the positions where the enzyme cuts its pattern (Hint: the location where the restriction site starts in a sequence). You can return a list containing name of the enzyme and the position it occurs in a dna sequence.
e.g. EcoRI=['gaattc', 1,154,...] [4 marks]

Continued...

QUESTION 3 [10 marks]

a) Write a python code to do complements and reverse complements for the given DNA sequence using Biopython packages (Seq, IUPAC)?
AGTAACCCTTAGCACTGGT

[4 marks]

b) By using suitable BioPython packages, write a Python code to open a FASTA file ("ls_orchid.fasta"), assign an iterator to the file, and print out "title" and "sequence" of the FASTA file. [4 marks]

c) Rewrite the previous question to read a GenBank file("input.gb"). Print out "organism" and "taxonomy". [2 marks]

Continued...

QUESTION 4 [10 marks]

a) Write an R program to read a sequence from the user. The program counts the number of bases, "A","T","C", and "G" prints the length of the sequence, bases, and the number of occurrences. [4 marks]

b) Write an R script to open and read a FASTA file ("laccase.fasta ") and store the sequence in a variable (seq1). Translate the DNA sequence from the fasta file and save the translated sequence (AA sequence) in a new FASTA file ("Laccase2.fasta"). Finally calculate the CG content of the sequence. [5 marks]

c) Write R code to draw a histogram of table D$wg with graph title "Weight gain", x and y axis labels "Gain", and "Freq" and red color bars. [1 mark]

Continued...

Question 5 [10 marks]

a) Imagine you want to build a database over DNA sequences from various animal species. Each data item should contain the **sequence** and the **name of the species**, and the **Latin name** to each data item. Create a Seq class to hold all the information needed for each data item. Write a constructor for class Seq to initiate data objects. Convert the sequence to upper case bases before initiation in the constructor. (Python) [3 marks]

b) Write down 2 functions for Class Seq in part "a" to access the values of "name" and "sequence" variables (getName(..) and getSequence(…) to retrieve the values). Write setSequence(..) function to receive a new value and update object's sequence value and also it sets a new information as the length of the sequence. [3 marks]

c) Explain briefly what each of the following R functions does. [4 marks]

i) dnorm(x)
ii) grep(pattern, x , ignore.case=FALSE, fixed=FALSE)
iii) ls()
iv) assign("even",c(2,4,6,8,10))
v)s3 = rep(a, 5)
vi)seq(4,10,by=2)
vii) cex.lab
viii) str(object)

End of Paper